

System Programming

Project #1 Find the minimum number using multi-process

(Due date: 11/12 11:59pm, Hard Deadline, No Extension)

- Project Objective

- Makefile을 작성할 수 있다.
- 멀티 프로세스, IPC과 관련된 시스템 콜 등을 활용하여 프로그램을 작성할 수 있다.

- Project Description

멀티 프로세스를 활용하여 주어진 숫자 행렬에서 가장 작은 값을 찾는다.

작성 해야 하는 코드는 proj1.c 파일과 main.c 파일로 구성되어 있다. proj1.c에는 구현되어 있지 않은 함수들 (FIX ME로 표기 되어 있음)을 구현하고, main 코드의 경우 주어진 주석에 맞게 코드 작성을 진행한다.

0. 본 프로젝트에서 제공되는 스켈레톤 코드는 C언어 기반의 코드로 작성이 되어 있으므로, C언어로 작성 해야함. 그리고 제공된 헤더 파일의 경우 (.h 파일), 절대 수정하지 말아야 함. 제출 시에는 Makefile과 .c (proj1.c, main.c) 파일에 대해서만 제출해야 하며, 채점 시에도 두 파일에 대해서만 확인하고 채점을 할 예정임

1. Makefile 작성

제공된 소스 코드 파일을 컴파일하기 위한 Makefile을 아래의 조건들을 충족하도록 작성하시오.

- 1) shell에서 make 입력 시, 소스 코드가 모두 컴파일 되어야함
- 2) shell에서 make clean 입력 시, 컴파일 도중 생성된 *.o 파일과 실행 파일을 모두 삭제 해야함

2. 본 프로그램 작성

- 1) 프로그램 실행에 필요한 정보는 프로그램 실행 시 shell에서 프로그램의 인자 (Arguments)로써 입력 받는다. 입력 받는 정보는 아래와 같다.

첫 번째: 생성할 (정렬할) 숫자의 총 개수.

두 번째: 생성할 프로세스의 총 개수

- 2) Shared Memory를 생성한 뒤, 부모 프로세스는 Shared Memory 영역에 generate_input 함수를 활용하여 무작위 순서를 갖는 정수를 저장한다.

- 작성 해야 하는 함수 : sets_shared_memory()

- 3) 자식 프로세스는 shared memory 상의 정렬되지 않은 숫자들을 일정하게 나누어서, 각

자 범위에 대한 숫자들 중 가장 작은 값을 찾아야 한다.

- 작성 해야 하는 함수: `child_find_min_value()`

4) 자식 프로세스는 자신이 구한 가장 작은 값을 `shared memory` 영역에 저장한다. 이 때, `shared memory`에는 이미 주어진 숫자 배열이 저장되어 있기 때문에 숫자 배열 이후의 영역에 저장을 하도록 한다. 자식 프로세스가 저장하는 가장 작은 값은 자식 프로세스 아이디 순서대로 정렬 된다.

- 작성 해야 하는 함수: `put_value()`

5) 부모 프로세스는 모든 자식 프로세스가 종료될 때까지 기다린 후, 자원을 수거한 뒤, 자식 프로세스가 저장한 최소 값들을 탐색하고, 그 중 가장 작은 값을 찾는다.

- 작성 해야 하는 함수: `parent_find_min_value`

6) 부모 프로세스는 찾은 최소 값을 출력 하고, `Shared Memory`를 `unlink` 한다.

3. 주의할 점

- `proj1.h` 에 새로운 함수 등을 추가하지 말 것.
테스트를 위해서 `BILLION` 값을 수정하는 것은 가능, 하지만 `proj1.h` 파일은 받지 않음
- `proj1.c` 의 함수 `argument`를 수정하지 말 것.
- 작성한 코드를 외부에 공개하지 말 것 (발견 될 경우 0점 처리)
- 컴파일시 `-lrt` 옵션 추가할 것 예시) `gcc -o hello a.o b.o -lrt`
- WSL 환경에서는 `POSIX shared memory`를 활용하는 것을 추천함.

4. 제출방법

ecampus에 *.c (`proj1.c`, `main.c`) 파일과, `Makefile` 파일을 올려주시길 바랍니다.

5. 질문 관련

ecampus에 올려주세요.